

5. A_Five - Bones

Program Name: A_Five.java

Input File: a_five.dat

A typical set of dominoes has 28 "bones", ranging in value from 0 to 6 on each half of a piece. For example the blank piece would be represented by 00, and the double six, often called the "boxcar", is 66. The pieces in ascending sequence would be 00, 01, 02, 03...32, 33, 34, 35...54, 55, 56, and 66. Keep in mind that the order of the two halves of a bone are interchangeable, so, for example, a "35" bone would be considered the same as the "53" bone.

For this problem you are given five dominoes randomly positioned in ten places along a horizontal line, followed by five more dominoes that are guaranteed to fit the five blank positions. For example, the first sample data line below represents an opening board that looks like this, where xx means a blank spot to be filled with the five remaining "bones":

5		3		X		X		4		5		5		6		X		X		6		1		1		3		X		X		X		X		X		X	
---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--

with 66, 34, 25, 36, and 62 to fill the blank spots, with a final solution of:

5		3		3		4		4		5		5		6		6		6		6		1		1		3		3		6		6		2		2		5	
---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--	---	--

Remembering that the dominoes can be reversed, the five remaining "bones" could be listed as in the second data example below: 66 43 52 63 26

with the same expected resulting output of:

53 34 45 56 66 61 13 36 62 25

Input: Several data sets, each line consisting of integers with single space separation, the first five indicating the positions (in the range 1-10) of the five "bones" already positioned on the board, followed by five pairs of integers representing the pieces in those positions, and then the remaining five pairs indicating the pieces to fill the blank spots.

Output: The resulting board with all pieces filled in correctly. A unique and successful solution for each data set is guaranteed.

Sample input:

```
1 3 4 6 7 5 3 4 5 5 6 6 1 1 3 6 6 3 4 2 5 3 6 6 2
1 3 4 6 7 5 3 4 5 5 6 6 1 1 3 6 6 4 3 5 2 6 3 2 6
1 3 5 7 9 2 3 4 5 6 6 1 2 6 0 3 4 6 1 5 6 5 0 2 6
```

Sample output:

```
53 34 45 56 66 61 13 36 62 25
53 34 45 56 66 61 13 36 62 25
23 34 45 56 66 61 12 26 60 05
```